Report

Embeddings

The two variations of embeddings that I trained were Skipgram and CBOW. The library I used to train them was `gensim` using the `Word2Vec` module. There was only one difference in the hyperparameter between them (at least in the case of the library that I was using), which was that in the case of Skipgram, a parameter `sg` was set to ‘1’, but we don’t use the parameter `sg` in the case of CBOW since the default flavour of `Word2Vec` module is CBOW.

The result for my five queries for each of the models (two models which I trained and two pre-trained models) were:

First Query - “female”

```
Skip-gram results for 'female':
[('language', 0.883082389831543), ('california', 0.8828102946281433), ('17', 0.882073700428009), (

CBOW results for 'female':
[('father', 0.30642038583755493), ('falling', 0.3037261366844177), ('switzerland', 0.2808740139007

GloVe results for 'female':
[('male', 0.9464975595474243), ('woman', 0.721383273601532), ('women', 0.7136487364768982), (

Word2Vec Demo results for 'female':
[('male', 0.8405333161354065), ('Female', 0.715976357460022), ('females', 0.6656177639961243), ('
```

Second Query - “united - america + england”

```
Skip-gram results for 'united - america + england':
[('asian', 0.9537683129310608), ('from', 0.9531306624412537), ('netherlands', 0.952884435653686

CBOW results for 'united - america + england':
[(',', 0.5398125052452087), ('and', 0.5045539140701294), ('in', 0.5023168921470642), ('(', 0.484583

GloVe results for 'united - america + england':
[('scotland', 0.717082142829895), ('manchester', 0.7026776671409607), ('newcastle', 0.6976361274

Word2Vec Demo results for 'united - america + england':
[('unite', 0.4655642807483673), ('disunited', 0.45898306369781494), ('unified', 0.457131952047348
```

Third Query - “world - war + peace”

```
Skip-gram results for 'world - war + peace':
[('28', 0.9127264022827148), ('saint', 0.9118055105209351), ('6', 0.9087514877319336), ('curie', 0.90

CBOW results for 'world - war + peace':
[('saint', 0.32202690839767456), ('great', 0.31078457832336426), ('28', 0.3038862347602844), ('fe

GloVe results for 'world - war + peace':
[('international', 0.6425808668136597), ('progress', 0.6143269538879395), ('conference', 0.6142356

Word2Vec Demo results for 'world - war + peace':
[('theworld', 0.4744304418563843), ('globe', 0.4341238737106323), ('continent', 0.3923476934432
```

Fourth Query - “year”

```
Skip-gram results for 'year':
[('april', 0.9920482039451599), ('world', 0.9912915825843811), ('day', 0.9906471371650696), ('may',

CBOW results for 'year':
[('january', 0.43319621682167053), ('president', 0.29340142011642456), ('world', 0.2924515008926

GloVe results for 'year':
[('last', 0.9042333364486694), ('month', 0.8928924202919006), ('years', 0.876212477684021), ('mo

Word2Vec Demo results for 'year':
[('month', 0.7653313875198364), ('week', 0.6598175764083862), ('months', 0.5790778398513794), (
```

Observation about the result

I noticed that the skip-gram and CBOW’s results were extremely noisy as compared to the pre-trained models. We can clearly see, the skip-gram has 10 most similar words to “female” as ‘language’, ‘california’, 17, 4, etc. which doesn’t give us a clear indication. Similar output can be observed for the CBOW model. Whereas the pre-trained models had extremely better results with the output for the input “female” being ‘woman’, ‘women’, ‘male’, etc. We can observe similar results for all the different inputs. Occasionally we see some good outputs like CBOW outputs ‘january’ for ‘year’ which is kind of related and skip-gram and CBOW both output ‘saint’ for word - war + peace which is also related.

I was surprised by these readings since I didn’t think the outputs would be significantly different. I also asked my friends if they had this problem, but for certain reasons which even I don’t know, they said their outputs weren’t as messed up as mine. I tried finding the root cause of this difference but couldn’t actually pinpoint the reason for this difference in the result. Using lowercasing, removing punctuations and removing stopwords improved the results but it still wasn’t near what the pre-trained models were giving. To an extent that was expected since the pre-trained models are trained on far larger datasets and longer training times whereas we used a simpler version of the original dataset which had millions of lines.

Bias

I studied the WEAT bias test which detects biases in word embeddings by associating pairs of concepts to each other. I extended it by introducing new concept pairs (past, future) with (celebration, conflict).

The result for the test is given below:

```
Skip-gram Model:
{'query_name': 'past_terms and future_terms wrt celebration_terms and conflict_terms',
'result': 0.25956932703653957, 'weat': 0.25956932703653957,
'effect_size': -0.21725883070877586, 'p_value': nan}

CBOW Model:
{'query_name': 'past_terms and future_terms wrt celebration_terms and conflict_terms',
'result': 0.10514171048998833, 'weat': 0.10514171048998833,
'effect_size': 0.3624498899482517, 'p_value': nan}

GloVe Model:
{'query_name': 'past_terms and future_terms wrt celebration_terms and conflict_terms',
'result': 0.432674230635166, 'weat': 0.432674230635166,
'effect_size': 1.2706931430869033, 'p_value': nan}

Word2Vec Google News Model:
{'query_name': 'past_terms and future_terms wrt celebration_terms and conflict_terms',
'result': 0.28021122366189954, 'weat': 0.28021122366189954,
'effect_size': 1.0660663672711386, 'p_value': nan}
```

We could observe that the glove model was the most biased since the result value was the highest for it. In the case of both glove and word2vec demo model, the past was associated with conflicts meanwhile the future is associated with celebrations. This was the opposite in the case of the skip-gram and cbow models even though this can be faintly seen. The results for cnow and skipgram are extremely noise.

Additionally, while performing the test we had to filter certain words from our vocab:

```
# First concept pair: Time periods (Past vs. Future)
past_terms = ["history", "old", "previous", "ancient", "traditional"]
future_terms = ["new", "become", "change", "create", "development"]

# Second concept pair: Celebration vs. Conflict
celebration_terms = ["festival", "celebrate", "holiday", "birthday", "party"]
conflict_terms = ["war", "force", "battle", "fight", "conflict"]
```

This was done because I was consistently getting NaN as the output for skip-gram and cbow due these models not having these words in their vocabs. Again I might be repeating this but I consulted my peers and this was only the case for me, which I found extremely surprising. Maybe because I was re-using a preprocessing function from assignment 1 which removed many crucial words, but I highly doubt that is the case. In any case, I wasn't able to pinpoint the exact reason for this observation.

The consequence of these biases can be severe because they can promote existing stereotypes, make biased predictions, discriminate against a certain group of people(even if unintended) and even misinform people about certain topics. We can even relate this to our test, since the model said implied that past is mostly related to conflicts, it may perpetuate negative views of historical events and cultures. It may even do harm in the opposite case where people believe that the opinion of "future" being related to "celebrations" is true because they might blindly believe that regardless of their present actions, the consequence will be positive.

Classification

The results for my classification task is as follows:

```
Bag-of-Words Results - Accuracy: 0.7545, F1: 0.7953
Embedding Results - Accuracy: 0.7824, F1: 0.8088

Feature Dimensions:
BoW: 5000
Embeddings: 100
```

I used the glove pre-trained embedding model because in my opinion it works better than other models. I came to this conclusion while doing the assignments where I noticed that it was giving better results as compared to the other models.

The classification task was to distinguish between two categories: 'alt.atheism' and 'soc.religion.christian' from the 20 Newsgroups dataset.

I think that the glove models' ability to capture semantic relationship between words was the main reason for a better accuracy and higher f1-score despite having much less dimensions than its counterpart. I came to this conclusion because that is exactly the fundamental distinction between a format like BoW which solely relies on numerical data without context as compared to a word embeddings format.

This suggests that word embeddings are arguably better and efficient at text classifications as compared to a simple numeric format of classifying text.

Note:- For this task **I used the help of chatGPT**, because my previous dataset and model from the assignment wasn't very good (the reasons for which I have explained extensively in my previous report) and therefore relied on data available using a library as suggested by chatGPT. The code was primarily written by me, I just followed it's suggestion to use Newsgroups dataset and how to use it.

Reflection

As mentioned before, the main takeway form this assignment was that word embeddings are really powerful tools which can offer a more efficient and accurate way to classify text. Even objectively without running accuracy tests I could see, it has many more use cases than a simple numeric classifier like BoW for eg. it can be used to find similar words based on the “context” of the data. Obviously nothing is all-powerful and comes without its cons, similarly, the main drawback of word embeddings is that they’re biased in their predictions. This is because the dataset might contain sentences which people who have these biases wrote and the model gets trained on that. Therefore prioritizing context has a major drawback which is that the context itself might not be very empathetic towards a certain group of people.

The output given by my skipgram and cbow was highly unexpected. I had already talked to certain peers who had done this assignment and none of them encountered the problem where there skipgram was giving them such “noisy” responses. I thought this might be due to the fact that I’m reusing a tokenizer from previous assignments and it might be interfering with the data. I was lemmatizing the data, but soon realised that reducing the words to their base from might be counter-productive in this case. Although, even after I removed the lemmatization property, I didn’t see much difference in the data. I wasn’t able to pin-point the exact fault in this case.

I think the professor had foreseen this issue and suggesting students to use pre-trained models is an idea that I find extremely useful beyond words. If this wasn’t suggested by the professor as part of the assignments I wouldn’t have been able to actual get an intuition of how good word embeddings are. Since the pre-trained model performed as intended, I was able to gauge how better they are as compared to some of the previous text classification methods we’ve seen before in class.

I also had to learn some libraries like usage of `Word2Vec` in `gensim` and that was one of the points where I got stuck reading the documentation. Additionally, I tried reusing my BoW model and dataset from previous assignments, but spent too much time on configuration because the model itself wasn’t very well trained and was outputting buggy results which weren’t very useful. This was a big roadblock for me while doing the assignment. Finally, I resorted to asking chatGPT for just a nudge in the right direction (i.e. what online dataset to use and a good library for calculations). ChatGPT suggested I use news20group dataset and perform calculations using the famous sklearn library which I was already familiar with, but had to read some documentations to refresh my memory.